



## 【特許請求の範囲】

【請求項1】それぞれ命令のシーケンスを含む複数の処理を実行可能であるコンピュータのプロセッサと主メモリとの間に配置されたキャッシュメモリを作動する方法において、

前記キャッシュメモリを、それぞれキャッシュメモリに項目を保持するためのアドレス指定可能な複数の記憶位置を有する複数のキャッシュ区分に分割するステップと、

前記キャッシュ区分のどれが当該処理の実行で使用するための項目を保持するために使用されるべきであるかを識別する区分表示子を各処理に割り当てるステップと、前記プロセッサが現在の処理の実行中に主メモリからの項目を要求し、かつこの項目が前記キャッシュメモリに保持されていない場合に、その項目を主メモリからフェッチし、かつ識別されたキャッシュ区分のアドレス指定可能な複数の記憶位置の中の1つにロードするステップとを含むことを特徴とする、キャッシュメモリを作動する方法。

【請求項2】現在実行されている現在処理のための前記区分表示子をメモリに保持するステップを有することを特徴とする請求項1に記載の方法。

【請求項3】新しい処理が前記プロセッサによって実行されるべきある場合、この新しい処理に割り当てられた新しい区分表示子が前記メモリにロードされることを特徴とする請求項2に記載の方法。

【請求項4】前記現在処理のための区分表示子を保持するメモリが前記処理についての状態情報をも保持する処理状態メモリであることを特徴とする請求項2または3に記載の方法。

【請求項5】前記区分表示子が前記処理のためのグループ識別子に含まれ、前記グループ識別子が前記処理のためのアドレス空間を識別することを特徴とする請求項1、2または3に記載の方法。

【請求項6】前記プロセッサが仮想ページ番号およびラインインページ番号を含むアドレスを発し、かつアドレス変換バッファが前記仮想ページ番号を、主メモリをアクセスする実ページ番号に変換するために備えられ、前記アドレス変換バッファが、前記グループ識別子をも受け取り、それから前記現在処理のための区分表示子を得ることを特徴とする請求項5に記載の方法。

【請求項7】各キャッシュ区分のアドレス指定可能な記憶位置数が増減可能であることを特徴とする請求項1ないし6のいずれかに記載の方法。

【請求項8】それぞれが一連の命令を含み、現在実行されている現在処理のための区分表示子を保持する処理状態メモリを含んでいる複数の処理を実行するためのプロセッサと、

主メモリと、それぞれ処理の実行中のプロセッサによって使用するた

めに前記主メモリからフェッチされる項目を保持するアドレス指定可能な複数の記憶位置を含む一連のキャッシュ区分を有するキャッシュメモリと、

前記主メモリから項目をフェッチし、この項目を前記アドレス指定可能な記憶位置のキャッシュメモリにロードするように構成され、現在処理と関連して処理状態メモリに保持された区分表示子に依り前記項目をロードする前記アドレス指定可能な記憶位置の中の1つを選択するキャッシュ取り替え機構とを備えたことを特徴とするコンピュータシステム。

【請求項9】前記区分表示子が前記処理のためのアドレス空間を識別する各処理のためのグループ識別子に含まれていることを特徴とする請求項8に記載のコンピュータシステム。

【請求項10】前記プロセッサが、仮想ページ番号およびラインインページ番号を含むアドレスを発し、かつ前記システムが前記仮想ページ番号を、主メモリをアクセスする実ページ番号に変換するアドレス変換バッファを備え、前記アドレス変換バッファが、前記グループ識別子をも受け取り、それから前記現在処理のための区分表示子を得るように作動できることを特徴とする請求項9に記載のコンピュータシステム。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プロセッサとコンピュータの主メモリとの間で作動するキャッシュシステム、特に複数の並行処理を実行することができるプロセッサに関するものである。

【0002】

【従来の技術】当該技術分野で周知のように、キャッシュメモリは、ある種のデータおよびコードへのアクセス待ち時間を減少させ、このデータおよびコードのために使用されるメモリバンド幅を減少させるようにコンピュータシステムにおいて使用されている。キャッシュメモリは、メモリアクセスを遅らせ、統合し、再配列することができる。

【0003】キャッシュメモリは、コンピュータのプロセッサと主メモリとの間で作動する。プロセッサで実行するプロセスによって要求されるデータや命令は、この処理を実行している間、キャッシュに保持することができる。キャッシュへのアクセスは、通常、主メモリへのアクセスよりも非常に高速である。プロセッサがキャッシュメモリに必要なデータ項目あるいは命令を置かなければ、プロセッサは、それを引き出すために主メモリを直接アクセスし、要求されたデータ項目あるいは命令はキャッシュにロードされる。キャッシュメモリを使用し、取り替えをするためのいろいろな公知のシステムが存在する。

【0004】実時間システムのキャッシュを当てにするために、キャッシュの動作は予測できる必要がある。す

なわち、キャッシュにあることが予想される特定のデータ項目あるいは命令が実際そこにあることに関して適度の確実度がある必要がある。大部分の既存の取り替え機構は、要求されたデータ項目あるいは命令をキャッシュに入れようと通常、常に試みている。これを行うために、この機構はキャッシュから他のデータ項目あるいは命令を削除しなければならない。これは、後で使用するためにそこにあることが予想される、削除項目を生じ得る。これは、特に、多重タスクプロセッサ、あるいは割り込み処理あるいは他の予測不可能な処理を実効しなければならないプロセッサの場合である。

【0005】

【発明が解決しようとする課題】本発明の目的は、複数の並行処理を実行するプロセッサに対してより大きな予測可能性のあるキャッシング動作をもたらすキャッシュシステムを提供することにある。

【0006】

【課題を解決するための手段】これに関しては、並行処理は、共有プロセッサによるが必ずしも同時に実行されない処理であるとみなされる。すなわち、第1の処理は実行し始め、何らかの理由で割り込むことができる。それから、プロセッサは、第2の処理を実行し始めるが、第1の処理が再び、あるいは何らかの他のプロンプトに応じていつでも実行できるようになったときに、第2の処理にいつでも割り込みできるようになっている。これは処理ハンドラによって管理される。第1の処理に関連したデータや命令が、第2の処理が実行している間にキャッシュから追い出されることが重要である。反対に、第2の処理は、実行している間、キャッシュにアクセスすることができることが有用である。例えば、2つの処理、すなわち処理Aおよび処理Bが1つのCPUで同時に実行している図8に示された場合を考察する。処理Aは最初にスケジュールされ、それがCPUを保有している間、そのCPUは、データキャッシュをその専用のデータで完全に満たすことができ、処理Bのためのデータキャッシュに入れられたいかなるデータをも追い出す。それから、制御は処理Bのためにスワッピングし、そのときデータキャッシュの状態を無効にし、処理Aのデータの全てを取り除き、その専用のキャッシュに入れる。データキャッシュ状態のこのデータのピンポンは、並行処理との間で共有しており、しばしば性能上、有害である。

【0007】請求項1に係る発明は、それぞれ命令のシーケンスを含む複数の処理を実行可能であるコンピュータのプロセッサと主メモリとの間に配置されたキャッシュメモリを作動する方法において、キャッシュメモリを、それぞれキャッシュメモリに項目を保持するためのアドレス指定可能な複数の記憶位置を有する複数のキャッシュ区分に分割するステップと、キャッシュ区分のどれが当該処理の実行で使用するための項目を保持するた

めに使用されるべきであるかを識別する区分表示子を各処理に割り当てるステップと、プロセッサが現在の処理の実行中に主メモリからの項目を要求し、かつこの項目がキャッシュメモリに保持されていない場合に、その項目を主メモリからフェッチし、かつ識別されたキャッシュ区分のアドレス指定可能な複数の記憶位置の中の1つにロードするステップとを含むことを特徴とするものである。

【0008】区分表示子を各処理に割り当てることによって、プロセッサで同時に実行している処理は、各々の他のデータや命令をキャッシュメモリから追い出すことが防止される。すなわち、例えばプロセッサで実行している第1の処理に割り当てられたキャッシュ区分はその後の第2の処理によって上書きすることができない。その代わりに、第2の処理は、それに割り当てられたその専用のキャッシュ区分を有する。一旦第1の処理が完全に完了すると、第1の処理に割り当てられたキャッシュ区分はそれから他の処理に割り当てることができるように、区分表示子を処理に割り当てることを変えることができることはもちろん好ましい。

【0009】処理の要求に応じて、2つ以上のキャッシュ区分を処理に割り当てるか、あるいはキャッシュへの処理アクセスをともかく否定することができる。

【0010】本発明の実施の形態においては、実行されている現在処理のための区分表示子は、処理についての状態情報をも保持する処理状態メモリに保持されている。これは、ここではスレッド状態ワードレジスタと呼ばれている。新しい処理がプロセッサによって実行されるべきである場合、新しいスレッド状態ワードは、この処理に割り当てられた新しい区分表示子とともにメモリにロードされる。

【0011】区分表示子は、処理のためのグループ識別子の中に含むことができ、グループ識別子は処理のためのアドレス空間を識別する。仮想アドレス指定方式では、プロセッサは、仮想ページ番号およびページ番号のラインを含むアドレスを発し、アドレス変換バッファは、仮想ページ番号を主メモリをアドレス指定する実ページ番号に変換するために備えられている。このように、アドレス変換バッファは、グループ識別子も受け取り、処理に割り当てられた仮想アドレス空間に応じて現在処理のための区分表示子をそこから得ることができる。

【0012】アドレス指定された項目のラインインページ番号は、項目が置かれるべきキャッシュ区分内のアドレス記憶位置を識別するために使用することができる。すなわち、各キャッシュ区分は直接マッピングされる。ラインインページ番号として項目アドレスのエンドビットであるが、適当なビットのセットに他ならない全てを使用することは必要ないことは明らかである。これらは、通常、アドレスの最下位の終わりに近い。

【0013】1つあるいはそれ以上のキャッシュ区分を処理に割り当てることができる。

【0014】このシステムは、キャッシュメモリからの項目を、前記項目の主メモリのアドレスに従って、かつ項目がキャッシュメモリに保持されるキャッシュ区分に関係なくアクセスするキャッシュアクセス回路を含み得る。すなわち、区分表示子は、取り替えて使用されるだけで、ルックアップでは使用されない。このように、キャッシュされる項目は、たとえこのキャッシング後にこの区分が異なるアドレス空間に関連した処理に今すぐ割り当てられるとしても、その区分から引き出すことができる。

【0015】請求項8に係る発明は、それぞれが一連の命令を含み、現在実行されている現在処理のための区分表示子を保持する処理状態メモリを含んでいる複数の処理を実行するためのプロセッサと、主メモリと、それぞれ処理の実行中のプロセッサによって使用するために主メモリからフェッチされる項目を保持するアドレス指定可能な複数の記憶位置を含む一連のキャッシュ区分を有するキャッシュメモリと、主メモリから項目をフェッチし、この項目をアドレス指定可能な記憶位置のキャッシュメモリにロードするように構成され、現在処理と関連して処理状態メモリに保持された区分表示子に依り項目をロードするアドレス指定可能な記憶位置の中の1つを選択するキャッシュ取り替え機構とを備えたことを特徴とするコンピュータシステムを要旨とするものである。

【0016】各処理は、共有ページ番号内部に主メモリのアドレスに保持された2つ以上の命令シーケンスを含むことができる。キャッシュ区分は、各キャッシュ区分を主メモリの特定の処理のページ番号に関連付けることによって処理に割り当てることができる。これは、我々の以前の英国特許出願第9701960、8号明細書に記載されている。

【0017】代替例として、区分表示子は、スレッド状態ワードレジスタに保持し、キャッシュ取り替え機構に直接供給することができる。

【0018】各キャッシュ区分におけるアドレス指定可能な記憶位置の数は変更可能でありうる。さらに、ページ番号に対するキャッシュ番号の関連性は、変更可能でありうると同時に、これらのページ番号を使用するプロセスはプロセッサによって実行される。

【0019】以下に記載する発明の実施の形態は、他の区分に入れられる他のデータのページからのキャッシュラインから読み出すかあるいはこのキャッシュラインに書き込むことによって予想できない追い出しからキャッシュの内容を保護するキャッシュシステムを示している。この実施の形態は、キャッシュの内容を予測できるシステムも提供する。

【0020】

【発明の実施の形態】図1は、キャッシュシステムを組

み込むコンピュータのブロック図である。コンピュータは、主メモリ6から項目をアクセスするためのアドレスバス4および項目をCPU2に戻すためのデータバス8に接続されているCPU2を備えている。データバス8はここではデータバスと呼ばれているが、この項目がCPUによる実行のための実際のデータあるいは命令を構成しようが構成しまいが、これは主メモリ6からの項目の戻りのためであることが理解される。ここに記載されているシステムは命令キャッシュおよびデータキャッシュの両方で使用するのに適している。公知であるように、別個のデータキャッシュおよび命令キャッシュがあってもよいし、あるいはデータキャッシュおよび命令キャッシュは結合されてもよい。ここに記載されたコンピュータでは、アドレス指定方式はいわゆる仮想アドレス指定方式である。アドレスは、ページアドレスのライン4aおよび仮想ページアドレス4bに分割される。仮想ページアドレス4bはアドレス変換バッファ(TLB)10に供給される。ページアドレスのライン4aはルックアップ回路12に供給される。アドレス変換バッファ10は、仮想ページアドレス4bに基づいて変換された実ページアドレス14をルックアップ回路12に供給する。ルックアップ回路12は、アドレスバスおよびデータバス16、18を介してキャッシュアクセス回路20に接続されている。さらに、データバス18は、主メモリ6からのデータ項目あるいは命令のためののものであってもよい。キャッシュアクセス回路20は、アドレスバス24、データバス26およびキャッシュメモリ22のための置換情報を転送する制御バス28を介してキャッシュメモリ22に接続されている。取り替えエンジン30は、取り替えエンジンとキャッシュアクセス回路との間で置換情報、データ項目(あるいは命令)およびアドレスを転送する取り替えバス32を介してキャッシュアクセス回路20に接続されている。取り替えエンジン30はそれ自体主メモリ6に接続されている。

【0021】取り替えエンジン30は、アドレス変換バッファ10から、主メモリ6の項目の実ページアドレスおよびラインインページアドレスを含む全実アドレス34を受け取る。取り替えエンジン30は、4ビットバス上のアドレス変換バッファ10から区分表示子をも受け取る。区分表示子の機能は以下に記載されている通りである。

【0022】最後に、取り替えエンジン30は、以下により明確に記載されているようにルックアップ回路12で発生される、ライン38上のミス信号を受け取る。

【0023】ここに記載されているキャッシュメモリ22は、直接マッピングキャッシュである。すなわち、このキャッシュメモリ22は複数のアドレス指定可能な記憶位置を有し、各位置は、キャッシュの1行を構成する。各行は、主メモリからの項目およびこの項目の主メモリのアドレスを含んでいる。各行は、この行に記憶さ

れたデータ項目の主メモリのアドレスの最下位ビットを示す多数のビットによって構成される行アドレスによってアドレス指定可能である。例えば、8行を有するキャッシュメモリの場合、各行アドレスは、これらの行を独自に識別するための3ビットの長さである。例えば、キャッシュの第2行は行アドレス001を有するので、ビット001で終わるアドレスを有する主メモリからの任意のデータ項目を主メモリに保持することができる。明らかに、主メモリでは、多数のこのようなアドレスがあるので、可能性としてキャッシュメモリのこの行に保持される多数のデータ項目がある。もちろん、キャッシュメモリは、毎回この行に唯一のデータ項目を保持することができる。

【0024】次に、図1に示されたコンピュータシステムの動作について説明するが、あたかも区分表示子が存在しないようである。CPU2は、主メモリのアドレスを使用して主メモリ6からの項目を要求し、このアドレスをアドレスバス4上に伝送する。仮想ページ番号は、それを所定の仮想/実ページ変換アルゴリズムに従って実ページ番号14に変換するアドレス変換バッファ10に供給される。実ページ番号14は、CPU2によって伝送された最初のアドレスのページ番号のライン4aとともにルックアップ回路12に供給される。ラインインページアドレスは、キャッシュメモリ22をアドレス指定するためにキャッシュアクセス回路20によって使用される。ラインインページアドレスは、キャッシュメモリ22の行アドレスに等しい、メモリの主アドレスの(必ずしもエンドビットを含まない)最下位数ビットのセットを含んでいる。データ項目(あるいは命令)およびデータ項目(あるいは命令)のアドレスである、ページアドレスの行によって識別される行アドレスのキャッシュメモリの内容は、ルックアップ回路12に供給される。そこで、キャッシュメモリから引き出されたアドレスの実ページ番号は、アドレス変換バッファ10から供給された実ページ番号と比較される。これらのアドレスが一致するならば、ルックアップ回路はヒットを示し、このヒットによってキャッシュメモリのこの行に保持されたデータ項目はデータバス8に沿ってCPUに戻される。しかしながら、キャッシュメモリ22のアドレス指定された行に保持されたアドレスの実ページ番号がアドレス変換バッファ10から供給された実ページ番号に一致しないならば、ミス信号は取り替えエンジン30のライン38上に発生される。バス34上のアドレス変換バッファ10から供給される実アドレスを使用して、主メモリ6から正しい項目を引き出すことは取り替えエンジン30のタスクである。主メモリ6から1度フェッチされたデータ項目は、取り替えバス32を介してキャッシュアクセス回路20に供給され、主メモリのアドレスとともにキャッシュメモリ22にロードされる。データ項目そのものはデータバス8に沿ってCPUにも戻される

ので、CPUは実行し続けることができる。上記に概略されるような直接マッピングキャッシュメモリでは、主メモリ6から呼び出されるデータ項目およびそのアドレスは、データ項目がチェックするために最初にアクセスされた記憶位置にロードされることは明らかである。すなわち、データ項目は、主メモリのラインインページアドレスの最下位数ビットのセットに一致する行アドレスを有するデータ項目を受け入れることができる唯一の記憶位置に上書きされる。もちろん、キャッシュメモリに最初に記憶されたデータ項目のページ番号およびいまキャッシュメモリにロードされるべきデータ項目は異なっている。この「1対1のマッピング」はキャッシュの有用性を制限する。

【0025】より大きな柔軟性をキャッシュシステムに提供するために、nウェイ・セット・アソシアティブ・キャッシュが開発された。4ウェイ・セット・アソシアティブ・キャッシュは図2に示されている。キャッシュメモリは4つのバンクB1、B2、B3、B4に分割される。このバンクは、図2の1行に対して概略的に示されているように、一般に共有行アドレスによって行方向にアクセスされる。しかしながら、この行は4つのキャッシュエントリ、すなわち各バンクに対して1つのキャッシュエントリを含んでいる。バンクB1に対するキャッシュエントリは、バス26a上に出力され、バンクB2に対するキャッシュエントリは、バス26b上に出力され、バンクB3およびB4に対しても同様である。したがって、これは、1行アドレス(あるいはページアドレスの行)に対して4つのキャッシュエントリを可能にする。行がアドレス指定される度に、4つのキャッシュエントリが出力され、そのアドレスの実ページ番号は、アドレス変換バッファ10から供給された実ページ番号と比較され、どのエントリが正しいエントリであるかを決定する。キャッシュに試みられたアクセスの際にキャッシュミスがあるならば、取り替えエンジン30は、主メモリから要求された項目を引き出し、例えば、特定の項目がどれくらいの長さキャッシュに保持されているか、あるいはシステムの他のプログラムパラメータに基づいている取り替えアルゴリズムに従って、バンクの中の1つの正しい行に要求された項目をロードする。このような置換アルゴリズムは公知であり、ここではこれ以上の説明は省略する。

【0026】それにもかかわらず、nウェイ・セット・アソシアティブ・キャッシュ(ここでは、nはバンク数であり、図2の4に等しい)は、単一の直接マッピングシステムの改良であるかぎり、なお汎用性がなく、より重要なことはキャッシュの動作を適切に予測できないことである。

【0027】ここに記載されたシステムは、より汎用性のあるキャッシュ取り替えシステムによってキャッシュメモリのコンピュータ使用の最適化を可能にするキャッ

シュ分割機構を備えている。

【0028】図3は、図1のコンピュータを使用するCPU2の概略ブロック図である。CPU2は、メモリバス4を介してメモリをアドレス指定し、データバス8を介してデータおよび命令を引き出すことができるフェッチ回路17に接続されている実行回路15を備えている。汎用レジスタ7のセットは、プロセスを実行する際に使用するためのデータおよび命令を保持する実行回路15に接続されている。さらに、参照番号9、11および13によって示される特別レジスタのセットが備えられている。任意の数の専用レジスタがあってもよく、例として、レジスタ11は、現在実行されているコードのラインを識別する命令ポインタを保持する。さらに、特殊レジスタ9は、CPU2によって実行されているプロセスの状態を規定するスレッド状態ワードを保持する。実行回路15は、毎回1プロセスあるいは命令のシーケンスを実行することができる。しかしながら、実行回路15は、同様にこのプロセスに割り込み、第1のプロセスが実行し終える前に別のプロセスを実行し始める。実行回路15によって現在実行されるプロセスに割り込むことができる多くの理由がある。1つは、より高い優先順位の割り込み処理が直ちに実行されることである。もう一つは、実行されている処理が現在長い待ち時間でデータを待っているため、第1の処理がこのデータを待っている間、その後の処理を実行し始めることは実行回路にとってより効率的であることである。データが受け取

られた場合、第1のプロセスを実行するために再スケジュールすることができる。並行処理の実行は、それ自体は公知であり、処理ハンドラ19によって管理される。

【0029】各処理は、いわゆる制御の「スレッド」の下で実行される。スレッドは下記の状態を有する。すなわち、この処理において、スレッドがどこに進むかを示している命令ポインタ、この処理が次にどこに分岐するかを示すジャンプポインタ、直ちにアクセスできる値を含んでいる汎用レジスタ7のセット、仮想アドレス/物理アドレスのマッピング、仮想アドレスによりアクセス可能なメモリの内容、および、スレッドおよび任意の他の値によってこのような浮動小数点丸めモード、スレッドがカーネル特権を有しているかどうか等をアクセスできる制御レジスタ。

【0030】上記の状態のいくつかは、ここではスレッド状態ワードと呼ばれ、スレッド状態ワードレジスタ9に保持されている小さい値のセットで指定される。スレッド状態ワードは、特に下記についての情報を保持する。スレッドがカーネルモードにあるか否か、スレッドはどの仮想アドレス空間にアクセスできるか、浮動小数点フラグ、トラップイネーブルおよびトラップモード、デバッグ情報、およびトラップ最適化情報。

【0031】スレッド状態ワードのフォーマットは表Iに規定されている。

【表1】

表 I

名 前	ビット*	サイズ	説 明
TSV.FPFLAG	0~7	8	浮動小数点例外フラグ
TSV.FPTRAP	8~15	8	浮動小数点例外トラップ
TSV.FPMODE	16~19	4	浮動小数点モード
	20~31		予備
TSV.USER	32	1	カーネルモード(0) ユーザモード(1)
TSV.SINGLE	33	1	単一ステップモード
TSV.TLB	34	1	第1のレベルTLBミスハンドラ表示子
TSV.WATCH	35	1	イネーブルされた監視ポイント
TSV.ENABLE	36	1	トラップイネーブル
	37~47	11	予備
TSV.GROUP	48~55	8	グループ番号
	56~63		予備

【0032】表Iから分かるように、スレッド状態は、8ビットのグループ番号を含んでいる。これは、キャッシュ区分を割り当てる区分表示子を発生するために下記に記載されるように示される。

【0033】ここに記載されたシステムのアドレス変換バッファ10では、各TLBエントリは、仮想ページ番号、実ページ番号および情報シーケンスに関連付けた。

情報シーケンスは公知であり、ここにこれ以上説明されていない方法でメモリのアドレスについてのいろいろな情報を含んでいる。しかしながら、現在説明されているシステムによれば、情報シーケンスは、グループ番号および仮想ページ番号によって決まる区分表示子を発生する区分コードをさらに含んでいる。これは、図4に概略図で示されている。ここでは、VPは仮想ページ番号を

示し、RPは実ページ番号を示し、GNがグループ番号を示し、INFOが情報シーケンスを示している。記載されている実施の形態では、PIは4ビットの長さである。

【0034】したがって、情報シーケンスINFOのビット0～3は区分表示子を構成する。区分表示子は、データ項目がキャッシュメモリ22に最初にロードされる時にデータ項目を入れることができる区分に関する情報を示している。図2に示されたキャッシュ構造に関して、各区分はキャッシュの1バンクを構成することができる。区分表示子において、各ビットはバンクの中の1つを参照する。区分表示子のビットjの1の値は、このページのデータを区分jに入れることができないことを意味する。ビットjの0の値は、このページのデータを区分jに入れることができることを意味する。データは、区分表示子の2ビット以上に0を有することによって2つ以上の区分に入れることができる。全て0である区分表示子によって、データはキャッシュの任意の区分に入れることができる。全て1である区分表示子によって、任意のデータ項目は、キャッシュメモリにロードすることができない。これは、例えば、キャッシュの内容を“凍結”するために、例えば診断目的のために使用することができる。

【0035】図4に示された例では、区分表示子は、主メモリにこの実ページ番号を有するデータ項目の置換がバンクB1あるいはB3を使用することができず、バンクB2あるいはB4が使用可能であることを示している。

【0036】2つ以上のバンクをページに割り当てることは事実上可能である。この場合、ラインインページアドレスがキャッシュに対して行アドレスよりも多くのビットを有するならば、区分はkウェイ・セット・アソシアティブ・キャッシュとして動作する。ここで、k個の区分は1ページに割り当てられる。したがって、記載された例では、図4の処理はバンクB2およびB4を使用することができる。しかしながら、この処理はバンクB1およびB3を使用することができない。

【0037】区分情報は、キャッシュルックアップで使用されないで、キャッシュ置換あるいは取り替えにだけ使用される。このように、キャッシュアクセスは、キャッシュメモリのどこにも保持されるデータ項目を探索できるのに対して、置換は、このページアドレスのために許可された区分にデータを置換するだけである。

【0038】図5は、取り替えエンジン30の内容をより詳細に示している。取り替えバス32は、3つの別個のバス、データバス32a、アドレスバス32bおよび置換情報を伝達するバス32cとして図4に示されている。データバスおよびアドレスバス32aおよび32cは、メモリバス54を介して主メモリをアクセスするメモリアクセス回路50に供給される。置換情報は、実ア

ドレス34、バス36上の区分表示子PIおよびミス信号38も受け取る決定回路52に供給される。決定回路52は、主メモリにアクセスされたデータが置かれるべきキャッシュの適切な区分を決定する。

【0039】ここに記載されたキャッシュ分割機構は特に多重タスクCPUに有用である。多重タスクプロセッサは、2つ以上の処理を「同時に」実行することができる。実際には、プロセッサは処理の一部を実行し、この処理が何かの理由で、多分続けるためのデータあるいはスティミュラスの要求で停止されると、プロセッサは直ちに他の処理を実行し始める。したがって、プロセッサは、個別処理を続けるためのデータあるいはスティミュラスを待って中止することができるときさえ常に作動している。図6は、このような状態を概略図で示している。図6の左側には、プロセッサが異なるP1、P2、P3、P4を実行することを引き受けることができるシーケンスが示されている。これらの処理が、そのデータがメモリに保持されることを予想できる場合の図が、図5の右側に示されている。したがって、処理P1のためのデータはページ0に保持されている。処理P2のためのデータはページ1およびページ2に保持されている。処理P3およびP4のためのデータはページ3を共有する。この例では、プロセッサは、第1の処理のシーケンスP1、第1の処理のシーケンスP2、第2の処理のシーケンスP1、第2の処理のシーケンスP2を実行し、それから第1の処理のシーケンスP3を実行する。第2の処理のシーケンスP1が実行された場合、処理P1はプロセッサによって完全に実行された。従来のキャッシュシステムでは、一旦プロセッサが第1の処理のシーケンスP2を実行し始め、このようにページ1からのアクセスを要求すると、これらのラインのデータ項目および命令は0ページからの予め記憶された記憶項目および命令をキャッシュで置換することは容易に明らかである。しかしながら、第2の処理のシーケンスP1が実行されると、これらはまもなく再び要求することができる。

【0040】ここに記載されているキャッシュ分割機構は、これから生じる得るタイミング遅れおよび不正確さを避ける。図7は、プロセッサが処理P1を実行している間のキャッシュの分割およびプロセッサが実行P3等にスイッチする場合の分割の変化を示している。図6は、各々の場合に対するTLBキャッシュ区分表示子も示している。したがって、図5は、プロセッサが処理P1およびP2を実行している間の分割されたキャッシュを左側に示している。処理P1はキャッシュのバンクB1およびB2を使用することができるが、バンクB3およびB4を使用することはできない。反対に、処理P2はバンクB3およびB4を使用することができるが、バンクB1およびB2を使用することはできない。これはTLBエントリで分かる。すなわち、処理P1は、バンクB1およびB2をアクセスすることができるが、バン

クB3およびB4にアクセスできないキャッシュ区分表示子を有する。処理P2は、バンクB3およびB4にアクセスすることができるが、バンクB1およびB2にアクセスできないキャッシュ区分表示子を有する。処理P3は、キャッシュにアクセスすることを防止するキャッシュ区分表示子を有する。したがって、処理P3からのデータ項目をキャッシュにロードしようとするプロセッサによるいかなる試みも禁止される。しかしながら、前述の処理シーケンスに関して、理解できるように、プロセッサは、処理P1を実行し終えるまで、処理P3のいかなる部分も実行するつもりはないために、これは不利なことではない。プロセッサが何らかの理由でP3を実行しなければならないならば、唯一の不利な面は、プロセッサがダイレクトメモリからそのアクセスを行わなければならない、キャッシュを使用できないことである。

【0041】処理P1が実行し終えた場合、プロセッサは、キャッシュ区分表示子を変えることができるようにカーネルモードを要求することができる。これが行われる方法は、分割機構が実行される方法で決まる。前述の例の場合、区分コードは任意の他のTLBエントリと同様にTLBにセットすることができる。このように、区分コードはCPU2で実行するカーネルモードソフトウェアによって通常セットされる。しかしながら、ユーザは、キャッシュ区分が換えられるべきであることを要求することによって区分を変えることができる。この場合、CPU2は、要求を実行するためにカーネルモードに換え、それに応じてTLBエントリを換え、それからユーザモードに戻し、ユーザは継続することができる。したがって、ユーザは、キャッシュの分割動作を変えることができるので、これまで可能であったよりも非常に大きな柔軟性をもたらす。この変化は図6の右側に示されている。したがって、次にキャッシュ区分表示子は、処理P1がキャッシュをともかく使用することを防止するが、キャッシュ区分表示子がキャッシュのこれらのバンクにアクセスできるように処理P3およびP4のためのキャッシュ区分表示子を変えることによってバンクB1およびB2を処理P3およびP4に割り当てる。このように、プロセッサが処理P3を実行することを予想している場合、プロセッサは今やキャッシュ機構を有する。

【0042】したがって、並行処理がデータキャッシュから他の各データを追い出すことを防止するシステムが前述された。すなわち、この処理はデータキャッシュ区分をばらばらにするようにマッピングされる。これは、その独自の専用データキャッシュに各処理を有効的にもたらす。これは両方の処理に使用可能なデータキャッシュ空間量を減少させるが、それによってその動作は正確に予測するのが非常に容易になる。ここに記載されているシステムの結果は図9に示されている。

【0043】ここに記載されているシステムが特に有用

である他の領域は性能に重要なルーチンの実行にある。その性能がシステムの全性能に絶対的に重要である若干のルーチンがある。これの適切な例は、呼び出される場合、保証された（通常短い）時間の長さに効果を生じなければならない割り込みサービスルーチンであってもよい。これらの場合、キャッシュ区分は、これらの重要なルーチンに必要なデータおよびコードのためのデータキャッシュおよび命令キャッシュの両方にとっておかれる。したがって、命令キャッシュおよびデータキャッシュの残りは残りの処理の中で分配することができる。図10は、可能な配置を示している。図10では、データキャッシュの1つの2キロバイトおよび命令キャッシュの1つの4キロバイトを動作に重要な割り込みサービスルーチンのために取っておく例が示されている。

【0044】本発明が上記に詳述された実施の形態に限定されないことは明らかである。いくつかの特定の可能な変更は後述されるが、これは本発明の範囲内で可能である包括的な変更のリストではない。

【0045】上記の実施の形態では、アドレスバス4上のCPUによって発されるアドレスは、仮想ページ番号4bおよびラインインページ4aに分割される。しかしながら、本発明は、全仮想アドレスがCPUからキャッシュのためのルックアップ回路に送られている場合に使用することもできる。反対に、本発明は、CPUが実アドレスを直接にルックアップ回路に発する場合にも適用可能である。重要なことは、キャッシュ区分表示子が実行されている特定の処理と関連して提供されていることである。

【0046】上記の実施の形態では、単一のキャッシュアクセス回路20は、ルックアップおよび取り替えの両方に関してキャッシュをアクセスするように示されている。しかしながら、キャッシュに取り替えのための付加アクセスポートを装備することも可能なので、ルックアップおよび取り替えはキャッシュメモリ22のための異なるアクセスポートによって行われる。

【0047】上述の実施の形態では、取り替えエンジン30およびキャッシュアクセス回路20は個別ブロックで示されている。しかしながら、その機能を組み合わせるルックアップおよび取り替えの両方を実行する単一のキャッシュアクセス回路にすることも可能である。

【0048】下記は、特定の処理と関連して区分表示子を発生するのに可能な代替例を示すものである。

【0049】1つの代替例では、区分表示子は、スレッド状態ワードTSWに直接入れられる。記載されたスレッド状態ワードの場合、これは、TSWの現在予備の56～59を新しいフィールドTSW.PIに割り当てることによって行うことができる。それから、TSW.PIの値は、CPUから直接取り替えエンジン30に送られる。これは、TLB10からよりもむしろCPU2から直接区分表示子を取り替えエンジン30に接続するた



めに図1に示されたアーキテクチャの修正を必要とする。これを実施するために、新しいスレッド状態ワードTSWは実行される次のスレッドのためにロードされる場合、区分表示子PIが変えられる。これは、スレッド状態ワードのパラメータをセットする特定の設定命令で行うことができる。

【0050】他の実施の形態では、仮想アドレスを使用しないでグループ番号を区分表示子PIにマッピングするアドレス変換バッファ10にテーブルを備えることができる。これは、区分表示子をもたらずグループ番号によって索引付けられたテーブルあるいは一致グループのための区分表示子をもたらずグループ番号/区分表示子対を有するテーブルを備えることによって行うことができる。この場合、図1のアーキテクチャは変わらないが、異なるテーブルはアドレス変換バッファ10で必要である。これを実施する場合、区分表示子は、2つのオペランド、すなわち変えられる制御レジスタ番号およびその新しい値を有する“put”命令を使用して変えることができる。全ての制御レジスタは、この命令を使用してこのレジスタにアクセスするために使用することができる番号を割り当てられるので、グループ番号/区分表示子テーブルの各エントリは独自の制御レジスタ番号を有する。

【0051】

【発明の効果】本発明によれば、複数の並行処理を実行するプロセッサに対してより大きな予測可能性のあるキャッシング動作をもたずキャッシュシステムを提供することができる。

【図面の簡単な説明】

本発明をより完全に理解し、同様に本発明を実行できる方法を示すために、いま例として添付図面が参照され

る。

【図1】キャッシュシステムを組み込むコンピュータのブロック図である。

【図2】4ウェイ・セット・アソシアティブ・キャッシュを示す略図である。

【図3】図1のCPUのブロック図である。

【図4】アドレス変換バッファのエントリの一例である。

【図5】取り替えエンジンのブロック図である。

【図6】多重タスクプロセッサの動作を示す図である。

【図7】図6のシステムのためのキャッシング動作の変更を示す図である。

【図8】未分割キャッシュを示す図である。

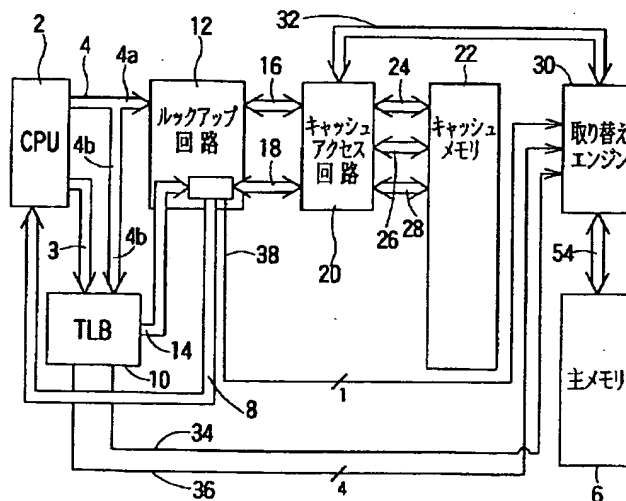
【図9】本発明の有用なアプリケーションを示す図である。

【図10】本発明の有用なアプリケーションを示す図である。

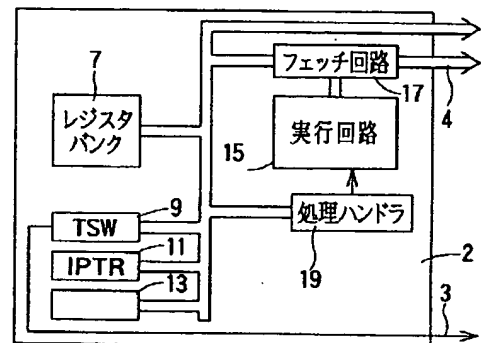
【符号の説明】

- 2 CPU
- 6 主メモリ
- 7 レジスタバンク
- 10 アドレス変換バッファ(TLB)
- 12 ルックアップ回路
- 15 実行回路
- 17 フェッチ回路
- 19 処理ハンドラ
- 20 キャッシュアクセス回路
- 22 キャッシュメモリ
- 30 取り替えエンジン
- 50 メモリアccess回路
- 52 決定回路

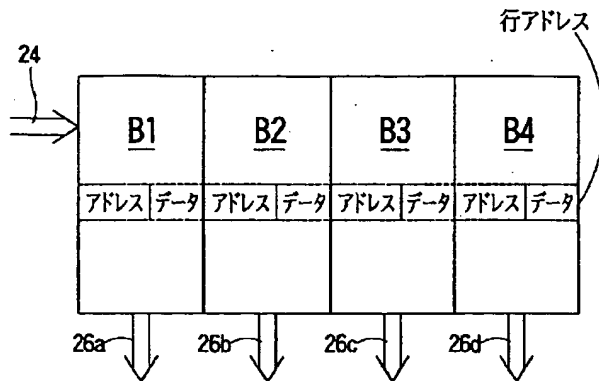
【図1】



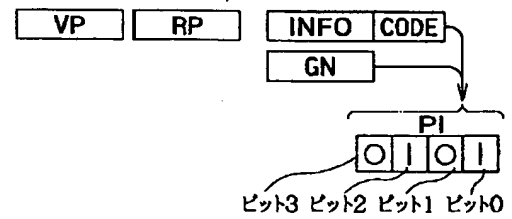
【図3】



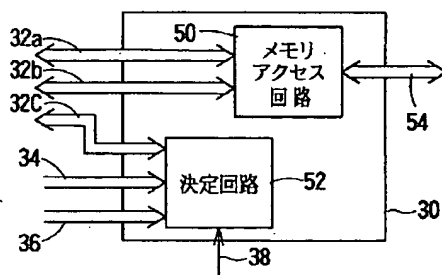
【図2】



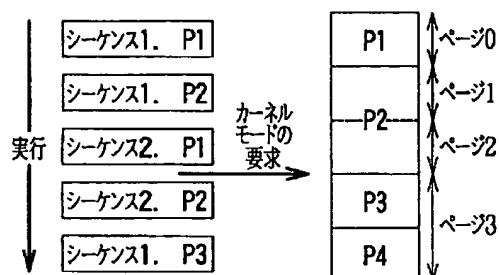
【図4】



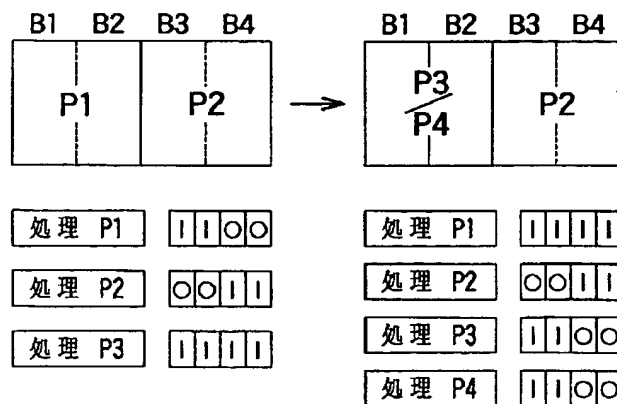
【図5】



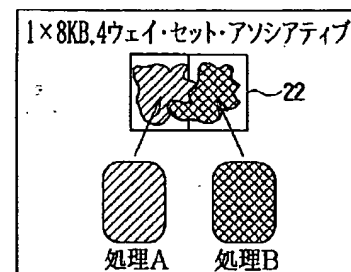
【図6】



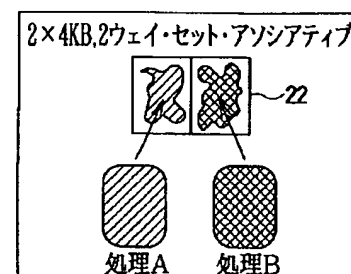
【図7】



【図8】



【図9】



【図10】

